



## **Trabalho em Grupo**

**Paradigmas de Programação | Diego Henrique Negretto**

**Tema: Fortran**

Bruna da Silva 104425

Caio Luís Goncalves 104294

Iago Henrique Honorato 104324

Isac Bruno 102507

Luiz Felipe Souza 102207

Wemerson Lucas 102158

## Sumário

Fortran .....	3
Compiladores .....	4
Versões .....	5
Ambiente da linguagem .....	6
Características e extensões .....	6
Constantes .....	7
Numéricas .....	7
Alfanuméricas .....	7
Maiúsculas e Minúsculas .....	7
Nomes de programas .....	7
Estrutura de um programa Fortran .....	8
Caracteres válidos em Fortran .....	8
Declaração de Variáveis .....	9
Especificador de tipo .....	9
Código - 'Hello World!' .....	10
Instalação da linguagem .....	10
Link GitHub .....	12
Bibliografia .....	13

## Fortran

A família de linguagem de programação Fortran foi desenvolvida a partir da década de 1950. O nome é um acrônimo da expressão "IBM Mathematical **FOR**mula **TRAN**slation System"

As versões iniciais da linguagem eram conhecidas como FORTRAN, mas a tipografia em caixa alta foi ignorada na representação das versões recentes da linguagem, começando a partir do Fortran 90.

A linguagem Fortran é principalmente usada em Ciência da computação e Análise numérica. Apesar de ter sido inicialmente uma linguagem de programação procedural, versões recentes de Fortran possuem características que permitem suportar programação orientada por objetos.

Fortran é uma linguagem compilada e pode se usar o compilador livre GCC.

O GCC (GNU Compiler Collection) é um conjunto de compiladores de linguagens de programação produzido pelo projeto GNU para construir um sistema operativo semelhante ao Unix livre.

## Compiladores

- GCC
- Gfortran
- G95
- F2c
- Intel Fortran Compiler
- IBM VisualAge
- Open64

O compilador g77 permite a compilação com Fortran77 até a versão 3.4.6.

Fortran90 e Fortran95 é usado o compilador a partir do GCC 4.0.0, que substitui g77 para GFortran.

O compilador g77, agora distribuído separadamente do GCC, pode ser incluído no GCC para compilar, e os seus binários são ainda disponibilizado para certos modelos de computadores.

A versão do FORTRAN IV desenvolvida na Universidade de Waterloo, Canadá, o compilador produzia melhores relatórios dos erros de compilação.

O software para a geração automática de fluxogramas a partir de programas em FORTRAN também foi desenvolvido fora da IBM.

## Versões

### Fortran 66 (1966):

Ampla disponibilidade de compiladores para diferentes computadores.

Simplicidade da linguagem.

Eficiência e as vantagens introduzidas pelo uso de sub-rotinas e compilação independente desta. Capacidade de lidar com número complexo.

### Fortran 77 (1978):

Facilitou a programação de estruturas com blocos.

« IF / THEN / ELSE / ENDIF »

Extensão introduzia entre outros « DO WHILE / END DO »

Característica marcante do FORTRAN 77 são as regras de alinhamento das linhas do programa.

### Fortran 90 (1990):

Foi considerada a versão final. Eficiente para os fins a que ela se destina. FORTRAN 90 trouxe uma novidade para o mundo das linguagens de programação: mecanismos para manipulação de arranjos.

### Fortran 95 (1995):

Revisão que acrescentou pequenas alterações ao FORTRAN 90, pela necessidade de aproximar o padrão à linguagem High Performance Fortran ou HPF a qual é voltada para o uso de computadores com arquiteturas avançadas. A linguagem HPF é uma espécie de extensão do FORTRAN 90 e, que muitos fabricantes oferecem compiladores HPF baseados na linguagem FORTRAN 90.

## Ambiente da linguagem

O Fortran roda em todos sistemas, existe duas formas disponíveis no mercado, a forma livre, uma versão com código aberto (open source) para sistemas operacionais Linux, e a versão paga (com licença proprietário).

### Características e extensões

O Fortran permite a criação de programas com velocidade de execução. Por isso, o uso em aplicações científicas como meteorologia, oceanografia, física, astronomia, geofísica, engenharia, economia etc.

Entre as linguagens especializadas mais populares baseadas em Fortran encontram-se a linguagem SAS, para gerar relatórios estatísticos, e SIMSCRIPT, para simulação de processos onde estão envolvidas listas de espera.

Foram adicionadas extensões à linguagem Fortran para fazer uso de características especiais do hardware, como: cache de instruções e pipeline da CPU. Por exemplo, um dos compiladores de Fortran da IBM (H Extended IUP) possuía um nível de otimização que reordenava as instruções do código para manter as unidades aritméticas da máquina ocupadas simultaneamente, obtendo assim um rendimento máximo. Estas extensões têm vindo desde então a desaparecer ou então os seus elementos têm sido incorporados no padrão principal da linguagem

Extensão da linguagem que ainda persiste é a OpenMP, que é uma extensão multiplataforma para programação de memória partilhada. Uma extensão recente, CoArray Fortran, tem como objetivo a promoção de programação paralela.

## Constantes

### Numéricas

Podem conter quaisquer valores reais, inteiros ou complexos. A parte decimal é separada da inteira por um ponto '.'. O expoente decimal é indicado pela letra 'e' ou 'E', deve vir entre o número e seu expoente sem o uso de espaços entre eles. Números negativos assim como a parte exponencial quando for negativa deve vir precedida do sinal menos '-'. Os números imaginários devem vir entre parênteses e a parte real deve ser separada por uma vírgula da parte imaginária.

### Alfanuméricas

São as 'strings', sequências de letras e/ou números. Podem conter qualquer sequência de caracteres não especiais. Devem vir entre aspas "" ou apóstrofos '. As aspas têm preferência sobre os apóstrofos. Não é permitido o uso de caracteres especiais e letras acentuadas.

### Maiúsculas e Minúsculas

FORTRAN não é 'case sensitive'. É permitido inclusive o uso do nome da variável escrita de formas diferentes num mesmo programa. EX.: VAR = var = Var.

### Nomes de programas

Os programas podem conter no início o seu nome (program nome\_do\_programa), e devem terminar com a palavra 'end' indicando que o que vem a seguir não precisa ser executado.

## Estrutura de um programa Fortran

- Cabecalho
- Conjunto de especificações
- Conjunto de instruções de execução
- Subprogramas (caso existam)
- Instrução de final do programa

## Caracteres válidos em Fortran

Caracter	Significado	Caracter	Significado
0, ..., 9	Algarismos	:	dois pontos
A, ..., Z	letras maiúsculas	=	sinal de igual
a, ..., z	letras minúsculas	!	ponto de exclamação
'	aplica ou apostrofe	&	e comercial
"	aspas	\$	sinal de dólar
(	parêntesis à esquerda	;	ponto e vírgula
)	parêntesis à direita	<	menor do que
*	asterisco	>	maior do que
+	sinal mais	%	símbolo de percentagem
-	sinal menos	?	ponto de interrogação
/	barra ( <i>slash</i> )	,	vírgula
	espaço em branco	.	ponto

Figura 1. Caracteres válidos na linguagem.

- Comentários (para se iniciar um comentário deve-se colocar o sinal !)
- Se uma instrução não couber numa linha, pode-se usar linhas de continuação (máximo de 39), onde o sinal `&` indica a continuação da instrução.



## Declaração de Variáveis

A linguagem Fortran define cinco tipos básicos de dados: · Inteiros – números sem parte decimal; podem ser positivos, negativos ou zero. · Reais – números com parte decimal; podem ser positivos, negativos ou zero. · Complexos – números com parte real e parte imaginária; qualquer das partes pode ser um inteiro ou um real. · Caracteres – sequência de caracteres ("strings") · Lógicos – valores lógicos (verdadeiro ou falso).

Para a declaração de variáveis simples: **especificador de tipo :: lista de variáveis**

A lista de variáveis é a lista de identificadores das variáveis, separados por vírgulas.

Especificador de tipo:

- INTEGER: Elemento inteiro é um valor numérico sem parte decimal (positivo, negativo ou zero).
- REAL: Elemento real é um valor numérico com parte decimal (positivo, negativo ou zero).
- COMPLEX: Um elemento complexo é um valor numérico com parte real e parte imaginária.
- CHARACTER (especificador de tamanho): É uma sequência de símbolos (ASCII) válidos em Fortran. Compreendidos entre plicas (') ou aspas (").
- LOGICAL: Um elemento lógico pode ter dois valores: verdadeiro ou falso. Respectivamente: .TRUE. - .FALSE.

Exemplos:

INTEGER :: num\_valores, factorial, soma

REAL :: juros, distancia

COMPLEX :: z\_val

CHARACTER(LEN = 15) :: primeiro\_nome, ultimo\_nome

LOGICAL :: questão

## Código - 'Hello World!'

```
program hello

    print *, 'Hello World!'

end program hello
```

### ----- Instalação da linguagem -----

Para poder compilar a linguagem FORTRAN, é necessário fazer o download de um compilador. O compilador que usamos é o G95 e pode ser encontrado nesse link: <http://www.fortran.com/the-fortran-company-homepage/whats-new/g95-windows-download/>

Após baixar e instalar o compilador, o usuário precisará de algum editor de texto, podendo usar até mesmo o bloco de notas, se desejar.

### ----- **Primeiros passos** -----

Em FORTRAN sempre iniciamos um código nomeando-o. Pra isso usamos o comando "program" e colocamos o nome do programa na frente dele. Para facilitar o entendimento, darei o nome de 'hello' para o nosso programa, sendo assim, começamos o código desse jeito:

```
program hello
```

### ----- **Comando print** -----

Se queremos escrever algo na tela, precisamos usar o comando "print" seguido de um asterisco e uma vírgula.

```
print *,
```

Como queremos que o programa mostre a frase 'Hello World!', precisamos somente implementar essa frase no comando print:

```
print *, 'Hello World!'
```

Com esse comando nosso programa mostrará na tela a frase ou palavra seguida do comando 'print'.

### ----- **Encerrando o código** -----

Para finalizar, precisamos dizer ao nosso código que o programa finaliza logo após ele executar o comando 'print'. Fazemos isso usando o comando 'end program' e o nome do nosso programa na frente.

Como nosso programa se chama 'hello', o comando ficaria desse jeito:

```
end program hello
```

### ----- **Juntando as partes** -----

Agora precisamos juntar todos os comandos descritos a cima para termos um programa funcional que imprime a frase 'Hello World!':

```
program hello
    print *, 'Hello World!'
end program hello
```

Agora basta salvar seu programa.

### ----- **Salvando nosso programa** -----

A linguagem FORTRAN conta com várias versões, porém, a que utilizaremos no tutorial será a versão 95. Por isso, quando for salvar seu documento de texto, salve como .f95 para evitar conflitos na hora de compilar.

```
hello.f95
```

### ----- **Compilando e executando** -----

Para compilar nosso código, precisaremos abrir o Prompt de Comando no Windows e navegar até a pasta onde nosso programa se encontra. Chegando no local do arquivo, chamaremos o compilador para transformar o nosso código em um executável.

```
g95 hello.f95 -o "nome do seu executável"
```

O compilador irá criar um arquivo executável com o nome escolhido e após isso, basta só você chamar seu executável.

```
[Compilando o código]
g95 hello.f95 -o hello.exe
```

```
[Executando o código]
hello.exe
```

E pronto, você criou seu primeiro código em FORTRAN!

## Link GitHub

Bruna da Silva 104425 - <https://github.com/Magdaleixon>

Caio Luís Goncalves 104294 - <https://github.com/Kayrito>

Iago Henrique Honorato 104324 - <https://github.com/honorato96>

Isac Bruno 102507 - <https://github.com/ZodakGhost/mediafortran>

Luiz Felipe Souza 102207 - <https://github.com/Felipeson>

Wemerson Lucas 102158 - <https://github.com/wemerson1967>

## Bibliografia

<https://pt.wikipedia.org/wiki/Fortran>

[https://webcache.googleusercontent.com/search?q=cache:xecm4wJdrVMJ:https://pt.wikiversity.org/wiki/Introdu%C3%A7%C3%A3o\\_%2C%2A0s\\_Linguagens\\_de\\_Programa%C3%A7%C3%A3o/Fortran+&cd=1&hl=pt-BR&ct=clnk&gl=br](https://webcache.googleusercontent.com/search?q=cache:xecm4wJdrVMJ:https://pt.wikiversity.org/wiki/Introdu%C3%A7%C3%A3o_%2C%2A0s_Linguagens_de_Programa%C3%A7%C3%A3o/Fortran+&cd=1&hl=pt-BR&ct=clnk&gl=br)

Guia básico de programação em linguagem Fortran 77 e 90 – PDF:

<https://www.inf.ufes.br/~thomas/fortran/tutorials/helder/fortran.pdf>

<https://web.fe.up.pt/~aarh/pc/PC-capitulo3.pdf>